ENTITY

An **entity** can be a real-world object.

Entity type is basically a collection of **entities** that have the similar attributes.

ATTRIBUTES

Entities are represented by means of their properties, called **attributes**. All **attributes** have values.

For example, a student entity may have name, class, and age as **attributes**. There exists a domain or range of values that can be assigned to **attributes**.

Types of attributes

- Identifying attributes(identifier):

  The attribute that is used to uniquely identify an instance of an entity .

  eg:Register No

- Descriptive attributes(descriptor):

  Non unique characteristics of an entity.

  eg:-name &age

- Simple attribute(atomic):

  The attributes that are indivisible .

  eg:age,price

- Composite attributes:

  The attributes that can be divided into smaller sub parts

  eg: address(It can be further divided into house

  name,place,district,pin,state.)

- Stored and derived attributes:

  Two or more attributes values are related in such a way that value of one attribute can be determined from the value of othet attribute.

  eg:Age can be determined from current date and value of DOB.Thus the attribute age is a derived attribute and DOB is known as stored attribute.

- Single valued attributes:

  The attribute that can have only one value for given entity.

  eg; book_titile.(One book can have only one title)

- Multivalued attribute:

  The attribute that can have multiple values for given entity.
  Eg:email-id,phone no(a student can have zero,one or more email id

  and phone no)

Null values for the attributes:

Sometimes there may be a situation where a particular entity may not have an appropriate value for an attribute.

Eg :Consider an entity type STUDENT with three attributes firstname,middlename,lastname.Since,not every student has a middlename,will have null value for the attribute middlename.

KEY ATTRIBUTES:

The attribute or combination of attributes whose values are distinct for each individual instance of an entity type is known as a key attribute.

An entity that has a key attribute is called strong entity type ,and doesn't have any key attribute is called weak entity type.

PRIMARY KEY

- The attribute used to uniquely identify an entity is primary key.
- A primary key is a field in a table which uniquely identifies each row/record in a database table.
- Primary keys must contain unique values.
- A primary key column cannot have NULL values. .
- When multiple fields are used as a primary key, they are called a composite key.

- **Foreign keys**

Foreign keys are the columns of a table that points to the **primary key** of **another table**. They act as a cross-reference between tables.

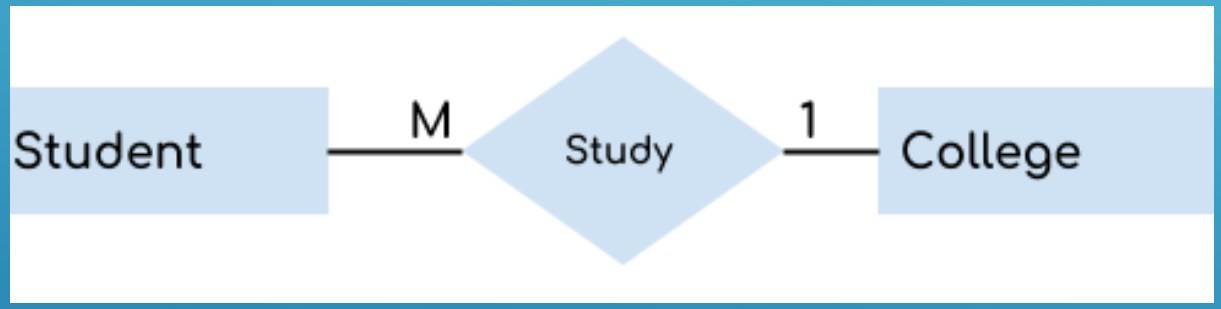
ENTITY RELATIOSHIP DIAGRAM(E-R Diagram)

- Describes the structure of a database with the help of a diagram, which is known as **Entity Relationship diagram.**
- An ER diagram shows the relationship among entity sets.
- An entity set is a group of similar entities and these entities can have attributes.
- ER diagram shows the complete logical structure of a database..

- **VARIOUS SYMBOLS USED ARE**
- **Rectangle**  : Represents Entity sets.
- **Ellipses**  : Attributes
- **Diamonds**  : Relationship Set
- **Lines**  : They link attributes to Entity Sets and Entity sets to Relationship Set
- **Double Ellipses**  : Multivalued Attributes
- **Dashed Ellipses**  : Derived Attributes
- **Double Rectangles**  : Weak Entity Sets
- **Double Lines**  : Total participation of an entity in relationship set

- ER Diagram has 3 main components
    1. Entity
    2. Attributes
    3. Relationship

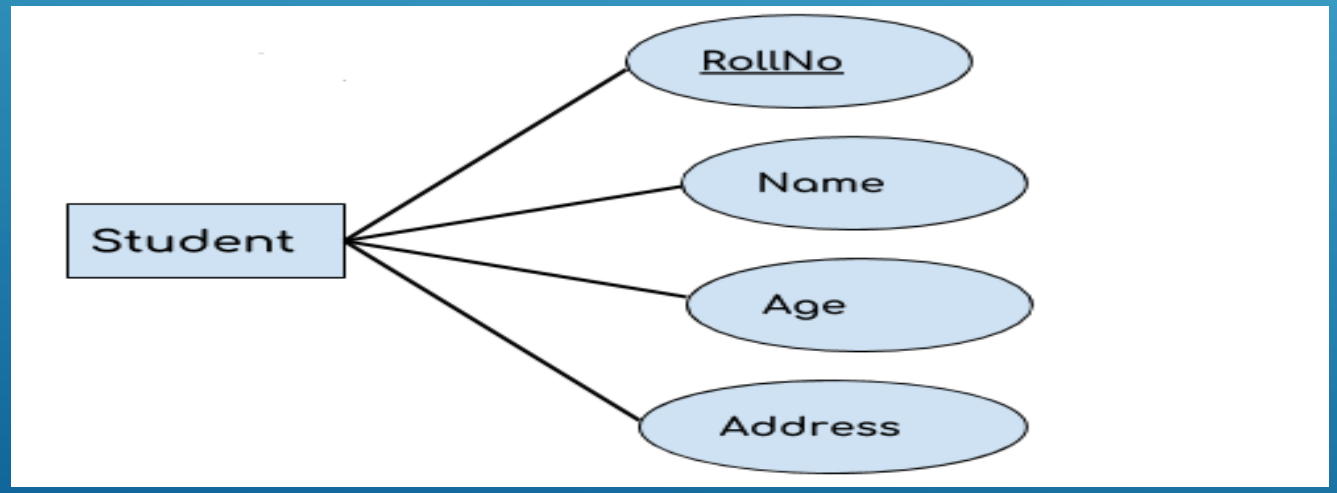
- **1. Entity**
- An entity is an object .
- An entity is represented as **rectangle** in an ER diagram.



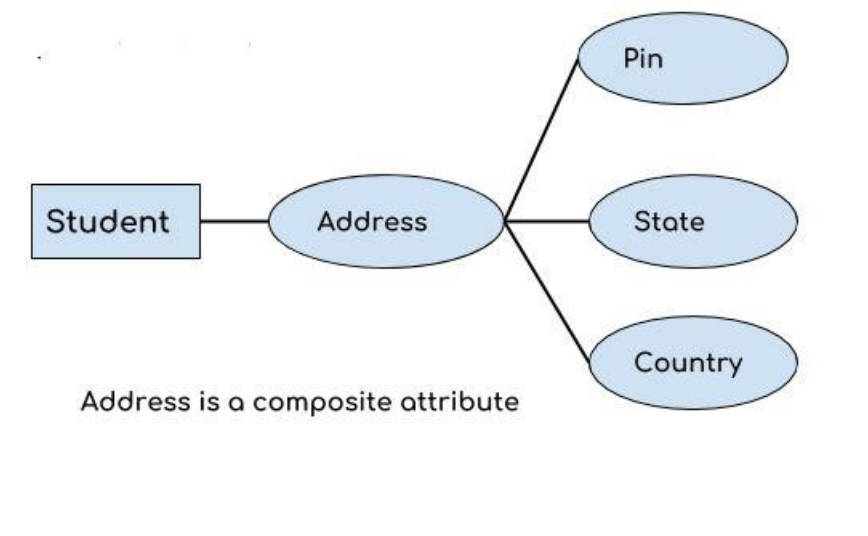
- For example: In the following ER diagram we have two entities Student and College and these two entities have many to one relationship as many students study in a single college.

- Weak Entity

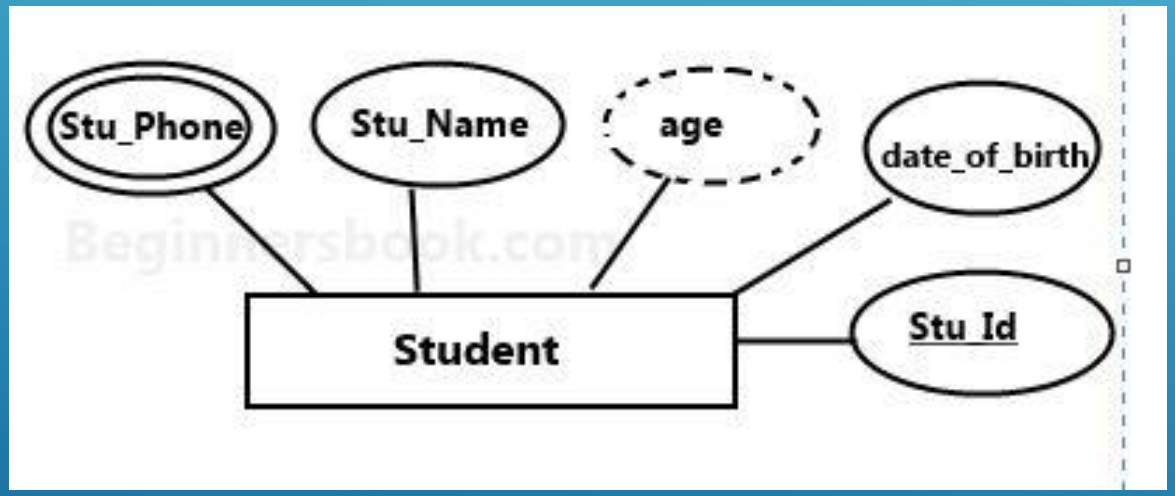
- **2. Attribute**
- An attribute describes the property of an entity. An attribute is represented as **Oval** in an ER diagram. There are four types of attributes:
- 1. Key attribute
  2. Composite attribute
  3. Multivalued attribute
  4. Derived attribute
- **Key attribute:**



- **2. Composite attribute:**



- **Multivalued attribute:**
- An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram
- **4. Derived attribute:**
- A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram.



- **Relationship**
- A relationship is represented by **diamond shape** in ER diagram, it shows the relationship among entities. There are four types of relationships:
  1. One to One
  2. One to Many
  3. Many to One
  4. Many to Many.

- **. One to One Relationship**
- When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is given to one person.
-
-

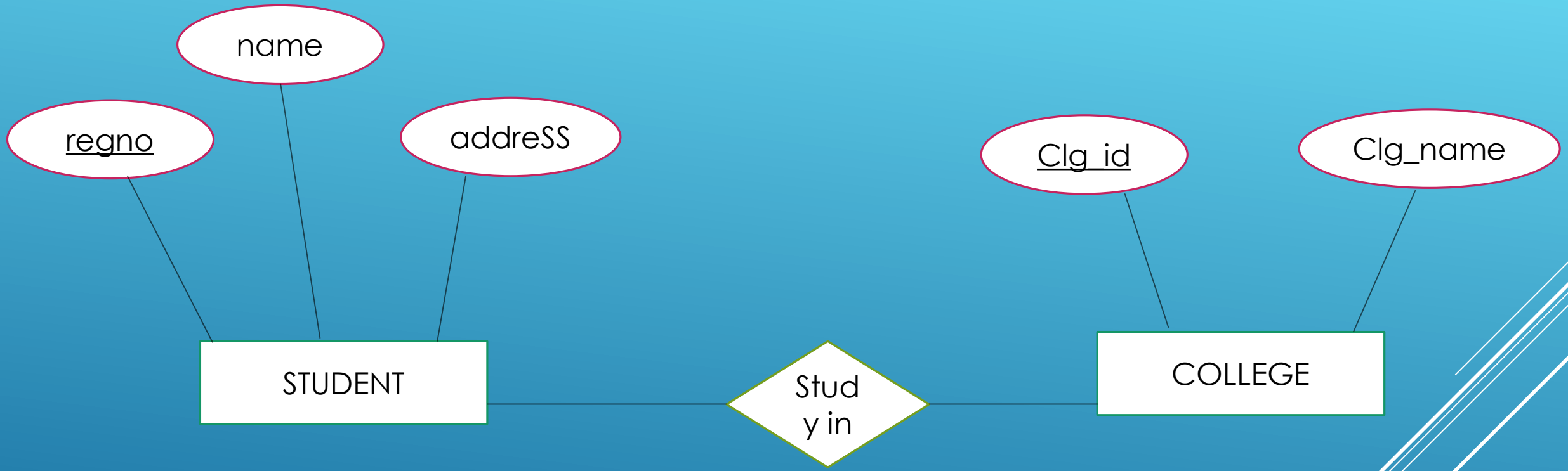
- **One to many**
- When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship.
- For example – a customer can place many orders but a order cannot be placed by many customers.
-

- **Many to One Relationship**
- When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship.
- For example – many students can study in a single college but a student cannot study in many colleges at the same time.
-

- **Many to Many Relationship**
- When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship.
- For example, a can be assigned to many projects and a project can be assigned to many students.
-

- **Total Participation of an Entity set**

- A Total participation of an entity set represents that each entity in entity set must have at least one relationship in a relationship set.


name
regno
addreSS
Clg_id
Clg_name
STUDENT
Stud
y in
COLLEGE

Enhanced E-R Model[EER]

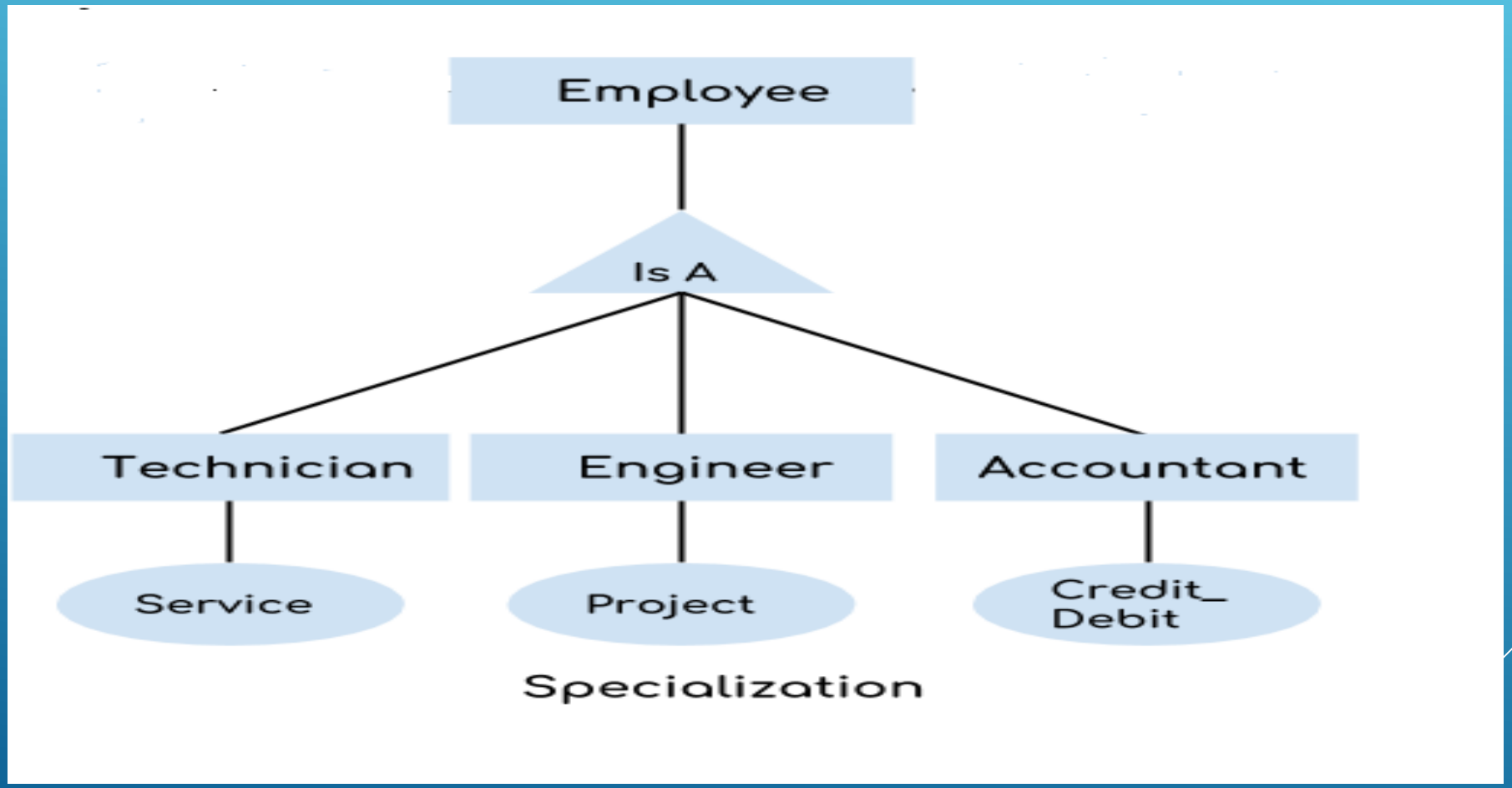
- It is used to represent the new and complex database applications such as telecommunications, Geographical information system.
- Extended E-R features are

  Specialization

  Generalization

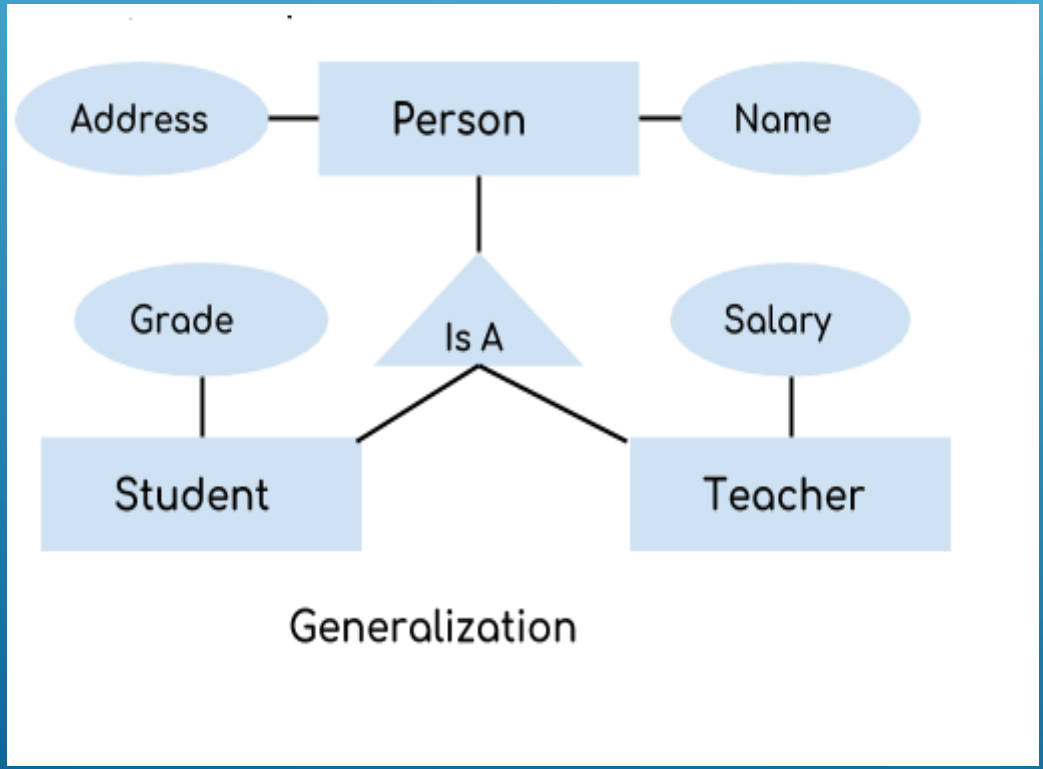
  Aggregation

- In **specialization**, a group of entities is divided into sub-groups based on their characteristics.
- Top-down approach.



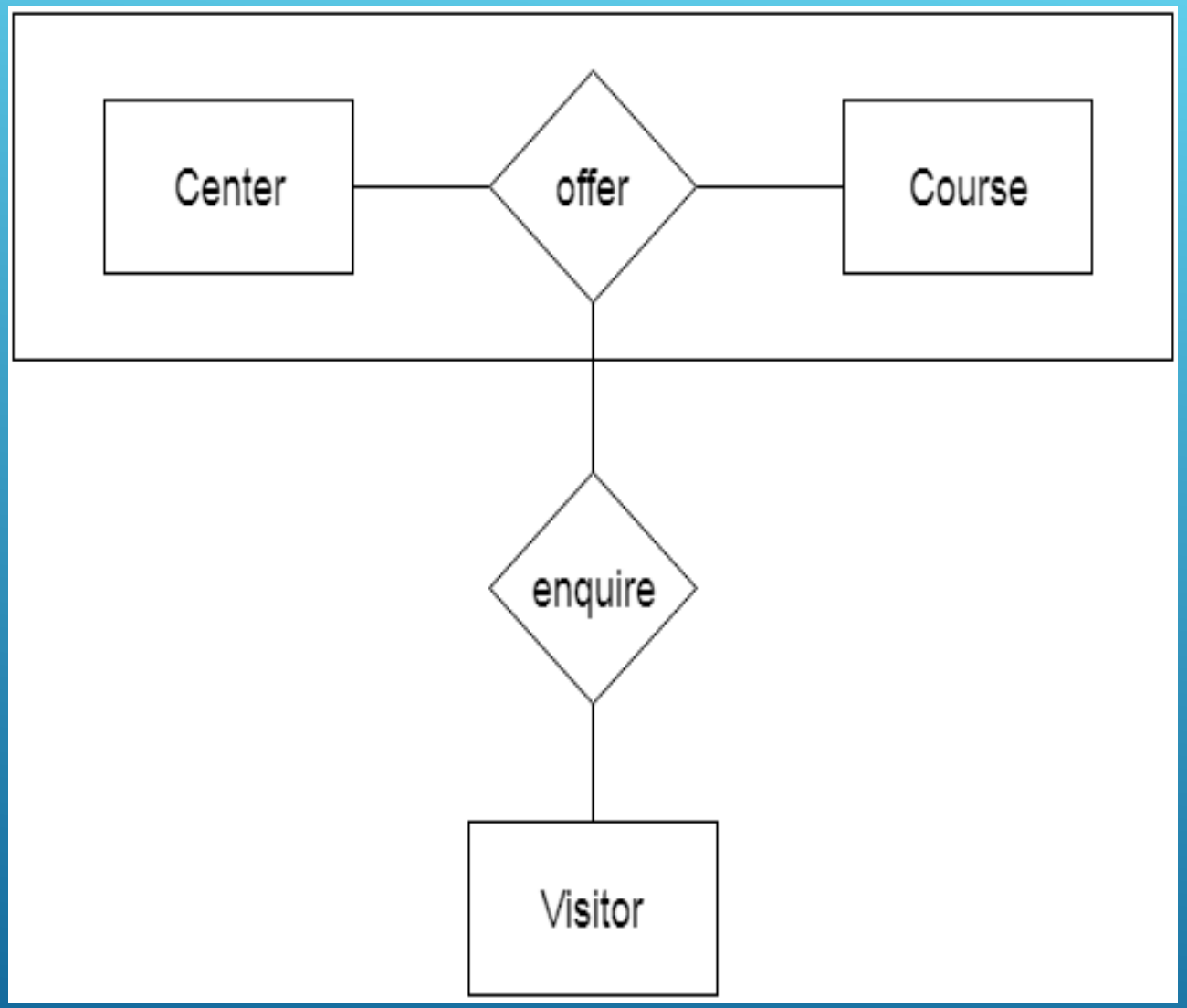
Specialization

- **In generalization**, a number of entities are brought together into one generalized entity based on their similar characteristics.
- Bottom-up approach.
- It is the reverse of specialization



Generalization

- In **aggregation**, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.
- For example: Center entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity visitor. In the real world, if a visitor visits a coaching center then he will never enquiry about the Course only or just about the Center instead he will ask the enquiry about both.


Center
offer
Course
enquire
Visitor

A super key (key) :it is a set of one or more attributes (columns), which can uniquely identify each record within a table.

Super key is a super set of candidate key.

Stud_ id	Name	Phone	Age
1	ASHNAD	1234567892	17
2	ASHNAD	4578136521	19
3	SAFA	4856723542	18
4	HASNA	2589631472	17
5	NIHAL	76184329823	18

Super keys :Stud _id,(studid,name),phone etc

Candidate key:

Super key without redundancy .

It is not reducible further.

Minimum (Single or combination of minimum attributes)

set of attributes used to uniquely differentiate record of the table.

ID	NAME	MARKS
S1	A	40
S2	B	20
S3	A	20
S4	C	30

HERE ID IS CANDIDATE KEY,BECAUSE ALL OTHER COLUMNS HAVING REDUNDANT DATA

ID	NAME	MARKS
S1	A	40
S2	A	40
S3	B	50
S2	B	50

ID-NAME :

S1A,

S2A

S3B

S2B SO (ID,NAME) CAN BE A CANDIDATE KEY,BECAUSE NO REDUNDANT DATA

NORMALIZATION

It is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updation anomalies. ... Normal forms are used to eliminate or reduce redundancy in **database tables**.

Here are the most commonly used normal forms:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce & Codd normal form (BCNF)

- **First normal form (1NF)**

As per the rule of first normal form, an attribute (column) of a table

cannot hold multiple values. It should hold only atomic values.

Emp-id	Emp-name	Emp-addree	Emp-mobile
101	SSS	Delhi	1234567891
102	bbb	Kerala	4578196833 1287564392
103	kkk	karnataka	2223355466 7894562863 1

- This table is **not in 1NF** as the rule says “each attribute of a table must have atomic (single) values”, the Emp_mobile values for employees bbb & kkk violates that rule.

Emp-id	Emp-name	Emp-addree	Emp-mobile
101	SSS	Delhi	1234567891
102	bbb	Kerala	4578196833
102	bbb	Kerala	1287564392
103	kkk	karnataka	2223355466
103	kkk	karnataka	78945628631

Second Normal Form (2NF)

A relation R is said to be in 2NF if

- Table is in INF
- Every **non-key** attribute in a table must be fully functionally depends on Primary Key

IDSTUD	LASTNAME	ID PROF	PROF	GRADE
1	CCC	3	VVV	5
2	AAA	2	GGG	4
3	BBB	1	SSS	6

IDSTUD	LASTNAME
1	CCC
2	AAA
3	BBB

ID PROF	PROF
3	VVV
2	GGG
1	SSS

IDSTUD	ID PROF	GRADE
1	3	5
2	2	4
3	1	6

NON-KEY ATTRIBUTE ‘GRADE’ DEPENDS UPON 2 OR MORE KEYS THEN IT IS CALLED FULLY FUNCTIONAL DEPENDENCY

- If A->B and B->C are two FDs then A->C is called transitive dependency.
- The normalization of 2NF relations to 3NF involves the removal of transitive dependencies. If a transitive dependency exists, we remove the transitively dependent attribute(s) from the relation by placing the attribute(s) in a new relation along with a copy of the determinant.

Third Normal Form (3NF)

A table design is said to be in 3NF if both the following conditions hold:

- Table must be in 2NF
- **Transitive functional dependency** of non-prime attribute should be removed.

emp_id	emp_name	emp_zip	emp_state	emp_city	emp_district
1001	John	282005	UP	Agra	Dayal Bagh
1002	Ajeet	222008	TN	Chennai	M-City
1006	Lora	282007	TN	Chennai	Urrapakkam
1101	Lilly	292008	UK	Pauri	Bhagwan
1201	Steve	222999	MP	Gwalior	Ratan

- Here, emp_state, emp_city & emp_district dependent on emp_zip.
- And, emp_zip is dependent on emp_id that makes non-prime attributes (emp_state, emp_city & emp_district) transitively dependent on emp_id. This violates the rule of 3NF.
- To make this table complies with 3NF we have to break the table into two tables to remove the transitive dependency:

emp_id	emp_name	emp_zip
1001	John	282005
1002	Ajeet	222008
1006	Lora	282007
1101	Lilly	292008
1201	Steve	222999

emp_zip	emp_state	emp_city	emp_district
282005	UP	Agra	Dayal Bagh
222008	TN	Chennai	M-City
282007	TN	Chennai	Urrapakkam
292008	UK	Pauri	Bhagwan
222999	MP	Gwalior	Ratan

BOYCE-CODD NORMAL FORM(BCNF) /3.5 NF

- Advanced Version of 3NF.
- It is in 3NF.
- For every FD X->Y, X should be super key of table.

STU	SUB	TEACHER
PINKI	DBMS	PRIYA
BHANU	DBMS	MADHU
DEEPU	JAVA	MADHU
JABBAR	DBMS	MADHU

- Here TEACHER is not a super key, but determines subject.
- Eliminate the redundant data.

STU	SUB
PINKI	DBMS
BHANU	DBMS
DEEPU	JAVA
JABBAR	DBMS

SUB	TEACHER
DBMS	PRIYA
DBMS	MADHU
JAVA	MADHU